Décodage en métrique rang et attaques sur un système de chiffrement à base de codes LRPC

Adrien Hauteville

Université de Limoges

Rapport de stage de MASTER 2

Auteur :
Adrien Hauteville

Encadrant :
Jean-Pierre Tillich

8 janvier 2015

Remerciements

Je tiens à remercier Jean-Pierre Tillich pour avoir accepté de me prendre en stage, pour ses idées et sa motivation, ainsi que toute l'équipe du projet SECRET pour son accueil chaleureux et sa bonne humeur.
Je remercie également Philippe Gaborit pour m'avoir permis d'obtenir ce stage et pour son soutien qu'il m'a apporté.

Table des matières

1 Introduction

Le chiffrement à clé publique et les protocoles de signature se fondent actuellement sur des problèmes mathématiques, essentiellement en théorie des nombres, réputés difficiles, tels que la factorisation (sur laquelle se base RSA) ou le problème du logarithme discret (cryptographie à base de courbes elliptiques). Ces problèmes ne sont pas à l'abri d'avancées en algorithmiques qui permettraient de les résoudre en temps raisonnable. Il a été récemment prouvé que le problème du logarithme discret sur des corps de petite caractéristique peut se résoudre en temps quasi-polynomial [1].

De plus, les protocoles basés sur ces problèmes sont facilement cassables par un ordinateur quantique (c'est-à-dire qu'il existe un algorithme résolvant ces problèmes en temps polynomial). Il est donc nécessaire de rechercher d'autres protocoles résistant à l'ordinateur quantique et basés sur d'autres problèmes. La cryptologie post-quantique est la branche de la cryptologie s'intéressant à ce type de problèmes, elle se fonde sur des problèmes difficiles (NP-complets ou NP-durs) d'algèbre linéaire, tels que le décodage des codes en métrique de Hamming, le décodage des codes en métrique rang [1] ou la recherche de vecteurs de petite taille dans un réseau euclidien.

La métrique rang présente l'avantage de fournir des cryptosystèmes avec une clé de faible taille pour une sécurité élevée (par exemple avec les LRPC [4]), avec un faible coût de chiffrement et de déchiffrement car les algorithmes n'utilisent que des opérations d'algèbre linéaire.

Dans ce stage, je me suis intéressé aux algorithmes de déchiffrement d'un code aléatoire en métrique rang, le but étant de chercher des analogies avec le déchiffrement en métrique de Hamming et si possible d'adapter les idées qui ont permis des avancées dans ce domaines à la métrique rang. J'ai aussi étudié une famille de codes, les LRPC[4], qui peuvent être utilisés en cryptographie post-quantique. Il est proposée une nouvelle attaque sur certains paramètres de ces codes.

1. plus précisément, il existe une réduction probabiliste [10] de ce problème à un problème NP-complet

2 Les codes

2.1 Introduction aux codes

Définition 2.1 (Code linéaire)**.** Un code linéaire $\mathcal{C}$ de longueur n et de dimension k sur un corps $\mathbb{F}_q$ (en général de caractéristique 2) est un sous-espace vectoriel de dimension k de $\mathbb{F}_q^n$, muni d'une fonction poids w telle que $\delta : (x, y) \mapsto w(x - y)$ soit une distance.
On dit que le code $\mathcal{C}$ est de type $[n, k]$ (ou plus simplement que $\mathcal{C}$ est une code $[n, k]$.

Le poids d'un mot $y \in \mathbb{F}_q^n$ est égal à $w(y) = \delta(y, 0)$. On peut ainsi définir la distance minimale d d'un code linéaire :

$$d = \min_{(x,y)\in\mathcal{C}^2} w(x - y) = \min_{x\in\mathcal{C}} w(x)$$

Afin de relier les messages, c'est-à-dire les éléments de $\mathbb{F}_q^k$, aux éléments du code $\mathcal{C}$, il est nécessaire de donner une représentation du code permettant d'utiliser un algorithme d'encodage. Pour cela, il faut introduire la notion de matrice génératrice.

Définition 2.2 (Matrice génératrice)**.** Soit $G \in \mathbb{F}_{q^m}^{k\times n}$ une matrice dont les lignes forment une base de $\mathcal{C}$. Cette matrice est appelé matrice génératrice du code.
L'encodage d'un message m (selon G) est :

$$c = mG$$

Définition 2.3 (Forme systématique)**.** Soient $\mathcal{C}$ un code $[n, k]$ en G une matrice génératrice de $\mathcal{C}$.
On dit que G est sous forme systématique sur les k premières positions si elle est de la forme :

$$G = \big(I_k|R\big) \text{ avec } R \in \mathbb{F}_q^{k\times(n-k)}$$

Sous cette forme, les éléments de $\mathcal{C}$ sont de la forme (m, mR) avec $m \in \mathbb{F}_q^k$.
À une permutation près, tout code possède une unique matrice génératrice sous forme systématique.

Définition 2.4 (Code dual)**.** Soit $\mathcal{C} = [n, k]$ un code linéaire sur $\mathbb{F}_q$. Le code dual de $\mathcal{C}$ noté $\mathcal{C}^\perp$ est l'ensemble :

$$\{x \in \mathbb{F}_q^n : \forall y \in \mathcal{C}, \langle x, y\rangle = \sum_{i=1}^{n} x_i y_i = 0\}$$

$\mathcal{C}^\perp$ est un code linéaire de type $[n, n - k]$.

Soit H une matrice génératrice de $\mathcal{C}^{\perp}$. Par définition, $c \in \mathcal{C} \Leftrightarrow Hc^T = 0$. La matrice H est appelée matrice de parité du code $\mathcal{C}$. Cette équivalence est extrêmement importante car elle donne un critère simple pour vérifier l'appartenance d'un mot à un code.

La théories des codes peut également s'appliquer au domaine de la cryptographie à clé publique. En 1978, McEliece propose un cryptosystème [8] basé sur le problème du décodage d'un code aléatoire en métrique de Hamming. Ce problème a été prouvé NP-dur la même année [2].
L'idée est de choisir un code dont on connaît un algorithme de décodage efficace puis de le masquer par un procédé algorithmique afin de le faire ressembler à un code aléatoire. La description de ce code apparemment aléatoire constitue la clé publique et la description du code utilisé pour le décodage (par exemple une matrice de parité de petit poids pour les MDPC ou bien le support et le polynôme de Goppa associé à un code de Goppa) la clé secrète.
Pour chiffrer un message m, Alice encode m et lui rajoute une erreur de poids r (r est un paramètre du cryptosystème).
Pour déchiffrer, la connaissance du code non masqué est nécessaire.

Voici plus explicitement le fonctionnement de l'algorithme :

clé publique : un code $\mathcal{C}$ de type $[n, k]$ sur $\mathbb{F}_q$, de matrice génératrice G apparemment aléatoire, dont on connaît un algorithme efficace de décodage. Un paramètre r désignant le poids des erreurs que l'on peut décoder.

clé secrète : une description de $\mathcal{C}$ permettant de décoder efficacement. Éventuellement une fonction secrète linéaire f permettant de retrouver le message à partir du mot décodé.

Entrées : m : message
G : matrice génératrice du code
Sorties : y : chiffré de m
Données : r : poids des erreurs que $\mathcal{C}$ peut corriger

début
  $e \leftarrow$ mot aléatoire de $\mathbb{F}_2^n$ de poids r ;
  $y \leftarrow mG + e$;
  retourner y
fin

Algorithme 1 : Chiffrement

Entrées : y : chiffré d'un message m
Sorties : m
Données : un algorithme $DECODER$ de décodage du code $\mathcal{C}$
la fonction linéaire f

début
  $m \leftarrow DECODER(y)$;
  $m \leftarrow f(m)$;
  retourner m
fin

Algorithme 2 : Déchiffrement

La sécurité du cryptosystème McEliece étant basée sur la difficulté de décoder un code aléatoire, il est important que l'attaquant ne puisse pas retrouver la structure sous-jacente du code, c'est pourquoi ce cryptosystème n'est utilisable qu'avec des familles de codes larges, pour résister aux attaques par Support Splitting [12].

2.2 Comparaison entre le métrique rang et la métrique de Hamming

Définition 2.5 (Métrique de Hamming)**.** Soit $\mathcal{C} = [n, k]$ un code linéaire. Le poids de Hamming d'un mot est le nombre de coordonnées non nulles :

$$w_h(c) = \#\{i \in [\![1; n]\!] : c_i \neq 0\}$$

En métrique de Hamming, le support d'une erreur de poids r est l'ensemble des r positions non nulles du vecteur. Ainsi, il y a $\binom{n}{r}$ supports possibles pour une erreur de poids r.

Définition 2.6 (Métrique rang)**.** Soit $\mathcal{C} = [n, k]$ un code linéaire à coefficients dans $\mathbb{F}_{q^m}$. Soit $(\beta_1 \dots \beta_n)$ une base de $\mathbb{F}_{q^m}$.

Tout élément $c \in \mathcal{C}$ peut être représenté de manière unique par une matrice $M \in \mathbb{F}_q^{m \times n}$ telle que :

$$\forall j \in [\![1; n]\!], c_j = \sum_{i=1}^{m} m_{ij}\beta_i$$

Le rang de c est le rang de la matrice M. Il est évident que ce rang ne dépend pas de la base choisie.

En métrique rang, le support d'une erreur de poids r est le sous-espace vectoriel de $\mathbb{F}_{q^m}$ de dimension r engendré par les coordonnées de l'erreur. Le nombre de supports possibles est donc égal au nombre de sous-espaces vectoriels de dimension r dans un espace de dimension m. Ce nombre que l'on note $\begin{bmatrix} m \\ r \end{bmatrix}_q$ est appelé coefficient de Gauss.

Proposition 2.7.

$$\begin{bmatrix} m \\ r \end{bmatrix}_q = \frac{(q^m-1)(q^m-q)\dots(q^m-q^{r-1})}{(q^r-1)(q^r-q)\dots(q^r-q^{r-1})} = \prod_{i=0}^{r-1} \frac{q^m-q^i}{q^r-q^i}$$

Démonstration. $(q^m-1)(q^m-q)\dots(q^m-q^{r-1})$ représente le nombre de familles libres de r vecteurs dans $\mathbb{F}_{q^m}$: en effet, on choisit le vecteur v_1 dans $\mathbb{F}_{q^m}\backslash\{0\}$ de cardinal q^m-1, puis v_2 dans $\mathbb{F}_{q^m}\backslash Vect(v_1)$ de cardinal q^m-q etc.
Deux familles $(\beta_1\dots\beta_r)$ et $(\gamma_1\dots\gamma_r)$ de r éléments engendrent le même espace vectoriel si et seulement si il existe une matrice $M \in GL_r(\mathbb{F}_q)$ telle que $(\beta_1\dots\beta_r) = M(\gamma_1\dots\gamma_r)$. Le nombre de matrices $r\times r$ inversibles est $(q^r-1)\dots(q^r-q^{r-1})$.
D'où $\begin{bmatrix} m \\ r \end{bmatrix}_q = \frac{(q^m-1)(q^m-q)\dots(q^m-q^{r-1})}{(q^r-1)(q^r-q)\dots(q^r-q^{r-1})}$ □

En approximant, il vient $\begin{bmatrix} m \\ r \end{bmatrix}_q \approx \frac{q^{rm}}{q^{r^2}} = q^{r(m-r)}$.

La linéarité par rapport à $\mathbb{F}_{q^m}$ des codes en métrique rang permet d'obtenir une description condensée de ces codes par rapport aux codes linéaires sur $\mathbb{F}_q$ de même paramètres. En effet, considérons un code $\mathcal{C}$ de type $[n,k]$ linéaire sur $\mathbb{F}_{q^m}$. Les mots de $\mathcal{C}$ sont de longueur nm sur $\mathbb{F}_q$ et les équations de parité sont au nombre de $(n-k)m$ sur $\mathbb{F}_q$.
Soit $\mathcal{C}'$ un code de type $[nm,km]$ sur $\mathbb{F}_q$. Ce code peut être également muni de la métrique rang en considérant les mots de $\mathcal{C}'$ comme des matrices de taille $m\times n$ sur $\mathbb{F}_q$. $\mathcal{C}'$ possède par ailleurs le même nombre d'équations de parité et la même longueur que $\mathcal{C}$ sur $\mathbb{F}_q$.
En revanche, $\mathcal{C}$ peut être représenté par une matrice de parité sous forme systématique de taille $(n-k)\times n$ à coefficients dans $\mathbb{F}_{q^m}$, soit une place mémoire de $k(n-k)m\lceil\log_2 q\rceil$ bits.
$\mathcal{C}'$ peut être représenté par une matrice de parité sous forme systématique de taille $(nm-km)\times nm$ à coefficients dans $\mathbb{F}_q$, soit une place mémoire de $km(nm-km)\lceil\log_2 q\rceil = k(n-k)m^2\lceil\log_2 q\rceil$ bits.
Ainsi, la linéarité par rapport à $\mathbb{F}_{q^m}$ permet de gagner un facteur m dans la taille des clés. Le gain en mémoire des codes en métrique rang par rapport à la métrique de Hamming est plus lié à ce facteur m qu'à la différence de taille des boules dans ces deux métriques.

2.3 Algorithme de décodage générique

Il est crucial de connaître précisément la complexité du problème du décodage en métrique rang, puisque la sécurité de nombreux cryptosystèmes et schémas de signature est déterminée par le coût du décodage d'un code aléatoire.

Un premier algorithme a été proposé par Chabaud et Stern en 1996 [3]. Pour décoder une erreur de poids r dans un code $[n, k]$ de corps de base $\mathbb{F}_{q^m}$, il est de complexité exponentielle de l'ordre de $q^{(m-r)(r-1)}$. Une nouvelle attaque a été trouvée par Ourivski et Johansson en 2003 [9] de facteur exponentiel $q^{(k+1)(r-1)}$. Il est intéressant de constater que ces deux algorithmes n'incluent pas la longueur n du code dans leur complexité. La longueur du code déterminant sa redondance, il est surprenant qu'un code plus long ne permette pas de décoder plus vite.

L'algorithme de Gaborit, Ruatta et Schrek[5] exploite pleinement la notion du support de l'erreur. Avant de présenter cet algorithme, introduisons certaines notations :

— le code est noté $\mathcal{C}$.
— sa longueur est n.
— sa dimension est k.
— m est la dimension du corps auxquels appartiennent les coefficients des mots du code. $\mathbb{F}_q$ est le corps de base.
— la matrice génératrice de $\mathcal{C}$ considérée est notée G, H est une matrice de parité.
— le mot bruité reçu est noté y et l'erreur est notée e : $y = mG + e$.
— le poids de l'erreur est noté r et son support est noté E.

L'idée est de rechercher un espace vectoriel E' de dimension $r' \geqslant r$ tel que les coordonnées de e appartiennent à E', c'est-à-dire que $E' \supset E$. En exprimant les coordonnées de e dans une base de E', il vient nr' inconnues.

Les équations de parité $He^T = Hy^T$ nous donne $(n-k)$ équations dans $\mathbb{F}_{q^m}$ soit $(n-k)m$ équations dans $\mathbb{F}_q$. Pour résoudre le système linéaire, il suffit d'avoir $nr' \leqslant (n-k)m$ donc $r' \leqslant \lfloor \frac{m(n-k)}{n} \rfloor = m + \lfloor \frac{-km}{n} \rfloor$.

Entrées : y : mot bruité
H : matrice de parité de $\mathcal{C}$
n, k, m : paramètres du code
Sorties : e : l'erreur calculée
Données : r : poids maximal de l'erreur e

début
  $s \leftarrow Hy^T$;
  $r' \leftarrow m + \lfloor \frac{-km}{n} \rfloor$;
  bool $\leftarrow$ TRUE;
  tant que *bool* **faire**
    choisir aléatoirement E' sous-espace vectoriel de dimension r';
    calculer une base $\{E'_1 \dots E'_{r'}\}$ de E';
    pour chaque $i \in [\![1; n]\!]$ **faire**
      exprimer e_i dans la base des (E'_j) ;
    finpour
    /* on obtient $r'n$ inconnues : $e_{ij}, i \in [\![1; n]\!], j \in [\![1; r']\!]$
       telles que $e_i = \sum_{j=1}^{r'} e_{ij} E'_j$ */
    résoudre le système $He^T = s$ d'inconnues e_{ij} dans $\mathbb{F}_q$;
    si *le poids de l'erreur obtenue est inférieur à* r **alors**
      bool $\leftarrow$ FALSE;
    finsi
  fintq
  retourner e
fin

Algorithme 3 : Décodage en métrique rang

Calculons la probabilité d'obtenir $E \subset E'$:
supposons que E' soit fixé.
Le nombre de choix pour E est le nombre de sous-espaces vectoriels de dimension r dans un espace de dimension m à coefficients dans $\mathbb{F}_q$, soit $\begin{bmatrix} m \\ r \end{bmatrix}_q$.
Le nombre de cas favorables est égal au nombre de sous-espaces vectoriels de E' de dimension r (il faut que E soit l'un d'entre eux), soit $\begin{bmatrix} r' \\ r \end{bmatrix}_q$.
Ainsi la probabilité recherchée vaut :

$$\mathbb{P}(E \subset E') = \frac{\begin{bmatrix} r' \\ r \end{bmatrix}_q}{\begin{bmatrix} m \\ r \end{bmatrix}_q} \approx \frac{q^{r(r'-r)}}{q^{r(m-r)}} = q^{-r(m-r')}$$

En prenant $r' = m + \lfloor \frac{-km}{n} \rfloor$, on obtient comme facteur exponentiel de l'algorithme $q^{-r\lfloor \frac{-km}{n} \rfloor} = q^{r\lceil \frac{km}{n} \rceil}$

Cet algorithme présente une analogie avec l'algorithme de décodage par ensembles d'information (Information Set Decoding ISD) en métrique de Hamming. En effet, cet algorithme utilise la propriété que l'on peut décoder un code $[n, k]$ en métrique de Hamming si l'on connaît un ensemble de k positions sans erreur. Autrement dit, cela revient à chercher un ensemble de $(n-k)$ positions contenant toutes les positions d'erreurs. Or la probabilité qu'un ensemble de $n-k$ positions contiennent les r positions d'erreur est $\frac{\binom{n-k}{r}}{\binom{n}{r}}$.

La structure des codes en métrique rang permet d'améliorer simplement le coût de l'algorithme. Au lieu de chercher directement l'erreur e, il est plus efficace de rechercher un multiple de e dans le code $\mathcal{C}'$ engendré par $\mathcal{C}$ et y. En effet, si r est inférieur à la capacité de correction de $\mathcal{C}$, alors les seuls mots de poids r de $\mathcal{C}'$ sont les multiples de e. Or il existe $\alpha \in \mathbb{F}_{q^m}$ tel que 1 appartienne au support de αe, on peut donc rechercher le support de l'erreur parmi les espaces vectoriels de dimension r' contenant 1.
Comme $\mathcal{C}'$ est un code de type $[n, k+1]$, les équations de parité donnent un système à $m(n-k-1)$ équations et nr' inconnues dans $\mathbb{F}_q$. Il suffit donc d'avoir $nr' \leqslant (n-k-1)m$ pour résoudre le système d'où $r' = m + \left\lfloor \frac{-(k+1)m}{n} \right\rfloor$.
Si r est supérieur ou égal à la capacité de correction de $\mathcal{C}$ alors une erreur de décodage peut survenir.

Remarque : Cet algorithme est en fait un algorithme de recherche de mots de petits poids dans un code aléatoire.

La probabilité d'obtenir $E \subset E'$ vaut :

$$\frac{\begin{bmatrix} r' \\ r-1 \end{bmatrix}_q}{\begin{bmatrix} m \\ r-1 \end{bmatrix}_q} \approx q^{-(r-1)(m-r')}$$

Ainsi, le facteur exponentiel de l'algorithme est $q^{(r-1)\lceil \frac{(k+1)m}{n} \rceil}$.
Comme k est beaucoup plus grand que r, le gain est important.

Il est important de noter que la connaissance de plus d'un vecteur d'une base du support de l'erreur permet de la même façon d'améliorer davantage l'algorithme : si l'on connaît a vecteurs du support, alors le facteur exponentiel est $q^{(r-a)\lceil \frac{km}{n} \rceil}$ pour un code $[n, k]$ sur $\mathbb{F}_{q^m}$. Nous montrerons par la suite que ce résultat peut s'appliquer pour l'algorithme d'Ourivski-Johansson [9], permettant d'obtenir un facteur exponentiel en $q^{(r-a)\lceil \frac{km}{n} \rceil}$.

Entrées : y : mot bruité
H : matrice de parité de $\mathcal{C}$
n, k, m : paramètres du code
Sorties : e : l'erreur calculée
Données : r : poids maximal de l'erreur e

début
  calculer une matrice de parité H' du code $\mathcal{C}'$;
  $r' \leftarrow m + \lfloor \frac{-(k+1)m}{n} \rfloor$;
  bool $\leftarrow$ TRUE;
  tant que *bool* **faire**
    choisir aléatoirement E' sous-espace vectoriel de dimension r' contenant 1;
    calculer une base $\{E'_1 \dots E'_{r'}\}$ de E';
    pour chaque $i \in [\![1; n]\!]$ **faire**
      exprimer e_i dans la base des (E'_j) ;
      /* $e_i = \sum_{j=1}^{r'} e_{ij} E'_j$ */
    finpour
    résoudre le système $H' e^T = 0$ d'inconnues e_{ij};
    si *le poids de l'erreur obtenue est inférieur à* r **alors**
      bool $\leftarrow$ FALSE;
    finsi
  fintq
  /* à ce stade on a calculé un multiple αe de l'erreur */
  calculer α à partir de l'égalité $H(\alpha e)^T = \alpha s$;
  retourner e
fin

Algorithme 4 : Décodage amélioré en métrique rang

2.4 Low Rank Parity Check codes

Les codes LRPC [4] sont une famille de code dont on connaît un algorithme de décodage en temps polynômial et possédant une faible structure algébrique, ce qui en font de bons candidats pour le cryposystème de McEliece.

Les codes LRPC sont des codes qui possèdent une matrice de parité dont tous les coefficients appartiennent au même sous-espace vectoriel de petite dimension. Ils sont donc semblables aux codes LDPC en métrique de Hamming.

Définition 2.8 (Codes LRPC)**.** Soit F un sous-espace vectoriel de $\mathbb{F}_{q^m}$ de dimension d.
Soit H une matrice $(n-k) \times n$ à coefficients dans F.
Le code $\mathcal{C}$ de matrice de parité H est un code LRPC de longueur n, de dimension k et de rang d.
La dimension de l'espace vectoriel engendré par les coefficients de H est au plus d. Cette dimension est appelée poids de H.

Un cas particulier des codes LRPC est le cas des codes LRPC quasi-cycliques (QC-LRPC).

Définition 2.9 (Codes LRPC quasi-cyclique)**.** Un code LRPC quasi-cyclique est un code LRPC possédant une matrice de parité H de petit poids d et quasi-cyclique.

Si la matrice H est doublement circulante, c'est-à-dire que H est la concaténation de deux matrices circulantes, on dit que le code LRPC engendré par H est doublement-circulant (LRPC-DC). Ces codes peuvent être utilisés dans le cryptosystème de McEliece et feront l'objet d'une attaque dans la quatrième partie.

Avant de présenter l'algorithme de décodage en temps polynomial des LRPC [4], il est nécessaire de rappeler quelques résultats sur le produit d'espaces vectoriels qui sont nécessaires pour justifier la correction de l'algorithme. Plus de précisions sont disponibles dans l'article introduisant les LRPC[4].

Définition 2.10 (Produit d'espaces vectoriels)**.** Soient A et B deux sous-espaces vectoriels de $\mathbb{F}_{q^m}$ de dimension respectives a et b.
L'espace vectoriel engendré par l'ensemble $\{xy : x \in A, y \in B\}$ est appelé produit des sous-espaces vectoriels A et B. Il est noté $\langle A.B \rangle$.

Soient $\{A_1 \dots A_a\}$ et $\{B_1 \dots B_b\}$ deux bases respectives de A et de B. Alors la famille $(A_i B_j)_{i \in [\![1;a]\!], j \in [\![1;b]\!]}$ est évidemment une famille génératrice de $\langle A.B \rangle$ donc $\dim \langle A.B \rangle \leqslant ab$.

La question qui se pose est de savoir avec quelle probabilité on a $\dim\langle A.B\rangle = ab$ dans le cas où $ab < m$.

Lemme 1 :[4] Soit A' et B deux sous-espaces de $\mathbb{F}_{q^m}$ de dimension respective a' et b tels que $\dim\langle A'.B\rangle = a'b$. Soit $A = A' + \langle\alpha\rangle$ où α est choisi uniformément aléatoirement dans $\mathbb{F}_{q^m}$. On a :

$$\mathbb{P}(\dim\langle A.B\rangle < (a'+1)b) \leqslant \frac{q^{(a'+1)b}}{q^m}$$

Proposition 2.11. *[4] Soit B un sous-espace vectoriel de dimension b et A le sous-espace vectoriel engendré par a éléments de $\mathbb{F}_{q^m}$ choisis aléatoirement de manière indépendante. Alors*

$$\mathbb{P}(\dim\langle A.B\rangle = ab) \geqslant 1 - a\frac{q^{ab}}{q^m}$$

Lemme 2 : Soient A et B deux sous-espace vectoriels de $\mathbb{F}_{q^m}$. Soient $a = \dim A$ et $b_2 = \dim\langle B^2\rangle$. Soit $e \in \langle A.B\rangle\backslash\{A\}$.
Supposons que $eB \subset \langle A.B\rangle$. Alors il existe $x \in B, x \notin \mathbb{F}_q$ tel que $xB \subset B$.

Proposition 2.12. *[4] Supposons que m soit premier. Soient A et B deux sous-espaces vectoriels de dimension respective a et b.*
Soient $(B_i)_{i\in[\![1;b]\!]}$ une base de B et $S = \langle A.B\rangle$. Alors :

$$\mathbb{P}(\bigcap_{i=1}^{b} B_i^{-1}S = A) \geqslant 1 - \frac{q^{ab(b-1)/2}}{q^m}$$

Proposition 2.13. *[4] Soit B un sous-espace de dimension b contenant 1 tel qu'il existe $\beta \in B$ tel que $\dim(B + b^{-1}B) = 2b-1$. Soit A un sous-espace choisi aléatoirement de dimension a. Alors :*

$$\mathbb{P}(\langle A.B\rangle \cap \beta^{-1}\langle A.B\rangle = A) \geqslant 1 - a\frac{q^{a(2b-1)}}{m}$$

Soit $\mathcal{C}$ un code LRPC de type $[n,k]$ sur $\mathbb{F}_{q^m}$ et de matrice de parité H de poids d.
Soit F le sous-espace engendré par les coefficients de H.
On suppose que l'on reçoit un mot y d'erreur e de poids r. Soit E le support de l'erreur de dimension r.
Soient $s = e^T H$ le syndrome associé à la matrice H et P le sous-espace vectoriel engendré par les coordonnées de s : $P = \langle s_1, \ldots, s_{n-k}\rangle$. Comme le poids d de la matrice H est petit devant k, la probabilité d'obtenir $P = \langle E.F\rangle$ est grande. L'idée est de retrouver le support de l'erreur à partir de la connaissance des espaces vectoriels F et P. Une fois le support de l'erreur connu, il est facile de calculer e en résolvant un système linéaire dont les inconnues sont les coordonnées des coefficients de e exprimées dans une base de E.

Entrées : y : mot bruité
H : matrice de parité de petit poids d
F : espace vectoriel de dimension d engendré par les coefficient de H
Sorties : e : l'erreur si le décodage s'est bien déroulé
ERREUR sinon
début
  $s \leftarrow Hy^T = (s_1 \dots s_{n-k})$;
  $S \leftarrow \langle s_1 \dots s_{n-k} \rangle$;
  /* $S = \langle E.F \rangle$ avec une forte probabilité */
  pour chaque $i \in [\![1; d]\!]$ **faire**
    $S_i \leftarrow F_i^{-1} S$;
    /* par hypothèse $S = \langle E.F \rangle$ donc $E \subset S_i$ */
  finpour
  $E \leftarrow \bigcap_{i=1}^{d} S_i$;
  /* ici E est le support de l'erreur avec une forte probabilité */
  résoudre le système $He^T = s$ en expriment les coordonnées de s et les équations dans la base $\{E_i F_j, i \in [\![1; r]\!], j \in [\![1; d]\!]\}$;
  /* le système a nr inconnues pour $(n-k)rd$ équations, on peut le résoudre à condition d'avoir $n \leqslant (n-k)d$ */
  si *l'erreur trouvée est de poids* r **alors**
    retourner e
  sinon
    retourner *ERREUR*
  finsi
fin

Algorithme 5 : Décodage d'un code LRPC

Les propositions précédentes permettent de choisir les paramètres du code pour s'assurer avec une très bonne probabilité deux conditions nécessaire à la réussite de l'algorithme : d'une part que $\dim\langle E.F \rangle = rd$ et d'autre part que l'intersection des S_i est égale à E.
Il reste à calculer la probabilité que l'espace engendré par les coordonnées du syndrome $s = He^T$ est bien égal à $\langle E.F \rangle$. Comme l'erreur est choisie aléatoirement, les coordonnées de s respectent aussi la distribution uniforme, il faut donc calculer la probabilité qu'un ensemble de $n-k$ vecteurs choisis aléatoirement dans un sous-espace vectoriel sur $\mathbb{F}_q$ de dimension rd engendre tout l'espace. En terme matricielle, cela revient à calculer la probabilité qu'une matrice $rd \times (n-k)$ à coefficients dans $\mathbb{F}_q$ soit de rang rd.

$$p = \frac{\#\{M \in \mathbb{F}_q^{rd \times (n-k)} : rang(M) = rd\}}{q^{(n-k)rd}} = \frac{\prod_{j=0}^{rd-1} q^{n-k} - q^j}{q^{rd(n-k)}}$$

On peut trouver une démonstration de cette formule dans ce livre [7].

2.4.1 Application des codes LRPC au cryptosystème McEliece

Soit $\mathcal{C}$ un code LRPC $[n, k]$ défini par une matrice de parité H $(n-k) \times n$ de poids d pouvant décoder des erreurs de poids r. La connaissance de H permet un décodage efficace, elle constitue la clé secrète du cryptosystème.
Pour masquer le code, on multiplie une matrice génératrice G de $\mathcal{C}$ une matrice inversible aléatoire R ou plus simplement donner G sous forme systématique. En effet, comme G et RG engendrent le même code, l'attaquant peut facilement calculer G_{syst} à partir de RG. Cela permet de réduire la taille des clés. La matrice G_{syst} est la clé publique du système.
Pour diminuer plus encore la taille des clés, on peut utiliser des codes $LRPC$ doublement circulants.
Des paramètres concrets de codes LRPC doublement circulants sont proposés dans cet article [4]. L'un de ces paramètres propose de choisir un code $[68, 34]$ sur $\mathbb{F}_{q^{23}}$ avec $q = 2^4$. Le choix d'une dimension non première offre une faille qui est exploitée dans la quatrième partie.

3 Amélioration du problème RSD

Dans cette section, nous nous intéresserons aux différentes améliorations qui peuvent être apportées au problème du décodage de codes aléatoires en métrique rang. L'algorithme proposé dans l'article[5] permet de définir rigoureusement le support de l'erreur en métrique rang et sa complexité s'apparente à la complexité de l'algorithme de décodage par ensemble d'information (ISD) en métrique de Hamming. Le but recherché est de transposer, si possible, les améliorations à l'algorithme ISD en métrique rang.

3.1 Brève introduction à l'ISD

Définition 3.1 (ensemble d'information)**.** Soit $\mathcal{C}$ un code $[n, k]$ sur $\mathbb{F}_q$. Soit G une matrice génératrice de $\mathcal{C}$. En distance de Hamming, un ensemble d'information I est un sous-ensemble de $[\![1; n]\!]$ de cardinal k tel que la restriction de G aux colonnes indicées par I soit inversible.

Le principe de l'algorithme ISD est de rechercher un ensemble d'information sur lequel la restriction de l'erreur est nulle.

Une première amélioration consiste à accepter que la restriction de l'erreur à l'ensemble d'information ne soit pas nulle, mais que son poids soit égal à un paramètre t. La probabilité de choisir un ensemble d'information avec un motif d'erreur de poids t est plus grande mais en contrepartie, il faut chercher tous les motifs de poids t dans cet ensemble d'information, ce qui a une complexité de l'ordre de $\binom{k}{t}$.

Pour améliorer cette recherche, une deuxième amélioration consiste à rechercher un motif d'erreur de poids t parmi un ensemble de $k+l$ éléments, où l est un autre paramètre de l'algorithme.

3.2 Application à la métrique rang

Dans leur article [5], les auteurs montrent comment la connaissance d'un espace vectoriel E' de dimension r' contenant le support E de l'erreur de poids r permet de décoder efficacement. Une première amélioration est de regarder la probabilité qu'un espace de dimension r choisi aléatoirement ait une intersection de dimension a avec un espace de dimension r' fixé.

Proposition 3.2. *Soit E' un sous-espace vectoriel fixé de $\mathbb{F}_q^m$ de dimension r'.*
Soit E un sous-espace vectoriel de $\mathbb{F}_q^m$ de dimension r choisi aléatoirement. La probabilité que l'intersection $E \cap E'$ soit de dimension a est donnée par

la formule :

$$\mathbb{P}(\dim E \cap E' = a) = \frac{\begin{bmatrix} r' \\ a \end{bmatrix}_q \begin{bmatrix} m-r' \\ r-a \end{bmatrix}_q q^{(r-a)(r'-a)}}{\begin{bmatrix} m \\ r \end{bmatrix}_q}$$

Pour démontrer cette proposition, nous allons procéder par étape et démontrer d'autres résultats intermédiaires intéressants. Introduisons d'abord quelques notations pour la suite de cette démonstration :

$\mathbf{E \leqslant F}$: E est un sous-espace vectoriel de F.

$\langle \mathbf{a_1} \dots \mathbf{a_n} \rangle$: espace vectoriel engendré par la famille $(a_1 \dots a_n)$

système de représentants : On suppose que tout un espace vectoriel quotient E/F est muni d'un système de représentants . Si $x \in E/F$, on notera son représentant $\dot{x}$ (ie $x = \dot{x} + F$)

#E : cardinal de E

Nous allons d'abord calculer le cardinal de l'ensemble des sous-espaces vectoriels de dimension r dont l'intersection avec E' est de dimension a :

$$f(a,r) = \#\{E \leqslant \mathbb{F}_q^m : \dim E = r \text{ et } \dim E \cap E' = a\}$$

Comme E' est fixé, la probabilité recherchée vaut $\dfrac{f(a,r)}{\begin{bmatrix} m \\ r \end{bmatrix}_q}$ où $\begin{bmatrix} m \\ r \end{bmatrix}_q$ est le nombre d'espaces vectoriels de dimension r.

Premièrement, regardons le nombre de choix possibles pour l'intersection $E \cap E'$, que nous appellerons H. H est un sous-espace vectoriel de E' de dimension a, il y a donc $\begin{bmatrix} r' \\ a \end{bmatrix}_q$ choix possibles pour H.

Deuxièmement, une fois l'intersection H fixée, il faut regarder le nombre de choix pour compléter H en un espace E de dimension r tel que $E \cap E' = H$. Pour cela, il faut étudier la projection de E sur l'espace quotient $\mathbb{F}_q^m/E'$.

Appelons W l'espace quotient $\mathbb{F}_q^m/E'$. On a donc $\dim W = m - r'$.
Soit φ le morphisme canonique de V dans W :

$$\begin{array}{rccc} \varphi & : & V & \to & W \\ & & x & \mapsto & x + E' \end{array}$$

Proposition 3.3. *Soit $E \leqslant V$ de dimension r.*
$\dim E \cap E' = a \Rightarrow \dim \varphi(E) = r - a$

Démonstration. Trivialement $\varphi(E)$ est un sous-espace vectoriel de W.
Soit $(e_1 \dots e_a)$ une base de $E \cap E'$.

On la complète en une base $(e_1 \dots e_r)$ de E
Trivialement $(\varphi(e_{a+1}) \dots \varphi(e_r))$ est une famille génératrice de $\varphi(E)$.
Soit $(\lambda_{a+1} \dots \lambda_r) \in \mathbb{F}_q^{r-a} : \sum_{i=a+1}^{r} \lambda_i \varphi(e_i) = 0_W$
$\Rightarrow \varphi(\sum_{i=a+1}^{r} \lambda_i e_i) = 0_W$
$\Rightarrow \sum_{i=a+1}^{r} \lambda_i e_i \in E'$
$\Rightarrow \sum_{i=a+1}^{r} \lambda_i e_i \in E' \cap E$
$\Rightarrow \sum_{i=a+1}^{r} \lambda_i e_i = 0$ par construction de la base de E
$\Rightarrow \forall i \in [\![a+1; r]\!], \lambda_i = 0$ donc la famille est libre.
D'où $\dim \varphi(E) = r - a$ □

À présent, montrons sa réciproque, c'est-à-dire que tout sous-espace de W de dimension $r - a$ est l'image par φ d'un espace E de dimension r tel que $E \cap E' = H$.

Proposition 3.4. *Soient $M \leqslant W$ de dimension $r - a$ et $H \leqslant E'$ de dimension a.*
Il existe $E \leqslant \mathbb{F}_q^m$ tel que :
- $\dim E = r$
- $\dim E \cap E' = H$
- $\varphi(E) = M$

Démonstration. Montrons d'abord le lemme suivant :

Lemme 1. Soit $(m_1 \dots m_n)$ une famille libre de M. Alors la famille de représentants $(\dot{m}_1 \dots \dot{m}_n)$ est libre dans $\mathbb{F}_q^m$ i.e.

$$\forall (\lambda_i)_{i \in [\![1;n]\!]} \in \mathbb{F}_q^n, \sum_{i=1}^{n} \lambda_i \dot{m}_i = 0 \Rightarrow \forall i \in [\![1; n]\!], \lambda_i = 0$$

Démonstration. $\sum_{i=1}^{n} \lambda_i \dot{m}_i = 0 \Rightarrow \varphi(\sum_{i=1}^{n} \lambda_i \dot{m}_i) = \sum_{i=1}^{n} \lambda_i m_i = 0$
$\Rightarrow \forall i \in [\![1; n]\!], \lambda_i = 0$ car la famille $(m_1 \dots m_n)$ est libre. □

Soit $(m_1 \dots m_{r-a})$ une base de M, alors la famille $(\dot{m}_1 \dots \dot{m}_{r-a})$ est libre.
Soit $(e_{r-a+1} \dots e_r)$ une base de H.
Montrons que $E = \langle \dot{m}_1 \dots \dot{m}_{r-a}, e_{r-a+1} \dots e_r \rangle$ vérifie bien les trois conditions.

— soit $(\lambda_1 \dots \lambda_r) \in \mathbb{F}_q^r$ tels que $\sum_{i=1}^{r-a} \lambda_i \dot{m}_i + \sum_{i=r-a+1}^{r} \lambda_i e_i = 0$.

En appliquant φ aux deux termes de l'égalité, il vient $\sum_{i=1}^{r-a} \lambda_i m_i = 0 \Rightarrow \lambda_i = 0$ pour tout $i \in [\![1; r-a]\!]$.

Donc $\sum_{i=r-a+1}^{r} \lambda_i e_i = 0 \Rightarrow \lambda_i = 0$ pour tout $i \in [\![r-a+1; r]\!]$.

D'où $\dim E = r$

— soit $x = \sum_{i=1}^{r-a} \lambda_i \dot{m}_i + \sum_{i=r-a+1}^{r} \lambda_i e_i \in E \cap E'$

$\Rightarrow \sum_{i=1}^{r-a} \lambda_i \dot{m}_i \in E'$

$\Rightarrow \sum_{i=1}^{r-a} \lambda_i m_i = 0$

$\Rightarrow x = \sum_{i=r-a+1}^{r} \lambda_i e_i \in H$

$H \subset E \cap E'$ est trivial.

— $\varphi(E) = M$ est trivial.

□

D'après cette proposition, une fois l'intersection H fixée, on a $\begin{bmatrix} m-r' \\ r-a \end{bmatrix}_q$ choix possibles pour $\varphi(E) = M$.
Une fois ces deux ensembles fixés, il faut encore dénombrer l'ensemble des espaces vectoriels $F \leqslant V$ tels que $\varphi(F) = M$ et $F \cap E' = H$.

Proposition 3.5. *Soient $M \leqslant W$ de dimension $r-a$ et $H \leqslant V$ de dimension a.*
$\#\{F \leqslant V : F \cap E' = H \text{ et } \varphi(F) = M\} = q^{(r-a)(r'-a)}$

Démonstration. Soient $(h_1 \dots h_a)$ une base de H et $(m_{a+1} \dots m_r)$ une base de M.
Soit $(x_{a+1} \dots x_r) \in (E'/H)^{r-a}$.
Soit $F = \langle h_1 \dots h_a, \dot{m}_{a+1} + \dot{x}_{a+1} \dots \dot{m}_r + \dot{x}_r \rangle$.
Trivialement, F vérifie bien les deux conditions de la proposition.

Montrons que tous les espaces de cette forme sont deux à deux distincts :

Soient $F = \langle h_1 \dots h_a, \dot{m}_{a+1} + \dot{x}_{a+1} \dots \dot{m}_r + \dot{x}_r \rangle$ et $F' = \langle h_1 \dots h_a, \dot{m}_{a+1} + \dot{x}'_{a+1} \dots \dot{m}_r + \dot{x}'_r \rangle$ tels que $(x_{a+1} \dots x_r) \neq (x'_{a+1} \dots x'_r)$

Quitte à réindexer, on peut supposer que $x'_r \neq x_r$
Par l'absurde, supposons que $F = F'$

$\Rightarrow \dot{m}_r + \dot{x}'_r = \sum_{i=1}^{a} \lambda_i e_i + \sum_{i=a+1}^{r} \lambda_i(\dot{m}_i + \dot{x}_i)$
$\Rightarrow \dot{m}_r - \sum_{i=a+1}^{r} \lambda_i \dot{m}_i = \sum_{i=1}^{a} \lambda_i e_i + (\sum_{i=a+1}^{r} \lambda \dot{x}_i) - \dot{x}'_r \in E'$
$\Rightarrow \lambda_r = 1$ et $\forall i \in [\![1; r-1]\!], \lambda_i = 0$ d'après le lemme 1.
D'où $\dot{x}_r = \dot{x}'_r$: contradiction.
Donc $F = F'$

Ainsi, le choix de F est déterminé par le choix du r-uplet $(x_{a+1} \dots x_r) \in (E'/H)^{r-a}$.
E'/H est de dimension $r' - a$ donc son cardinal vaut $q^{r'-a}$ donc le nombre d'espaces vectoriels vérifiant les deux conditions vaut $(q^{r'-a})^{r-a} = q^{(r'-a)(r-a)}$

□

Pour conclure :
— on a $\begin{bmatrix} r' \\ a \end{bmatrix}_q$ choix pour $H = E \cap E'$
— une fois H fixé, on a $\begin{bmatrix} m-r' \\ r-a \end{bmatrix}_q$ choix pour $M = \varphi(E)$
— enfin on a $q^{(r'-a)(r-a)}$ espaces vectoriels E vérifiant $\varphi(E) = M$ et $E \cap E' = H$

ce qui démontre que $\#\{E \leqslant V : \dim E = r \text{ et } \dim E \cap E' = a\} = \begin{bmatrix} m-r' \\ r-a \end{bmatrix}_q \begin{bmatrix} r' \\ a \end{bmatrix}_q q^{(r'-a)(r-a)}$
d'où le résultat annoncé.

À la différence de la métrique de Hamming, la notion de complémentaire d'un support n'est pas immédiate : en effet, en métrique de Hamming, si un ensemble d'information possède un motif de t erreurs, alors les $n-k$ autres positions contiennent $r-t$ erreurs.
En métrique rang, une première idée est d'écrire l'espace vectoriel $\mathbb{F}_q^m$ comme somme directe d'espaces vectoriels. Cela présente deux inconvénients, d'une part le supplémentaire n'est pas unique, d'autre part découper $\mathbb{F}_q^m$ en somme directe ne permet pas de séparer le support de l'erreur en deux espaces vectoriels dont la somme des poids est égale au poids de l'erreur.
Plus concrètement, soient E_1 et E_2 deux sous-espaces vectoriels de $\mathbb{F}_q^m$ tels que $\mathbb{F}_q^m = E_1 \oplus E_2$. Soit E le support de l'erreur de poids r. Notons $d_1 = \dim E_1 \cap E$ et $d_2 = \dim E_2 \cap E$. Alors $d_1 + d_2 \leqslant r$.

Une autre idée est de considérer l'espace vectoriel quotient comme complémentaire, ce qui permet d'avoir la propriété d'unicité.

Définition 3.6 (Complémentaire du support)**.** Soit $\mathcal{C}$ un code $[n, k]$ sur $\mathbb{F}_{q^m}$. Soit E le support d'un mot de poids r. E est un sous-espace vectoriel de $\mathbb{F}_q^m$

de dimension r. Le complémentaire de E est l'espace vectoriel quotient $\mathbb{F}_q^m/E$ de dimension $m-r$.

Soit φ la surjection canonique de $\mathbb{F}_q^m$ sur $\mathbb{F}_q^m/E$. D'après la proposition 2.12, on a la propriété que pour tout sous-espace vectoriel E' de $\mathbb{F}_q^m$ vérifie $\dim E \cap E' + \dim \varphi(E') = \dim(E')$. Cette propriété correspond à la même propriété du complémentaire en métrique de Hamming.
Ces idées sont un premier pas vers un algorithme améliorant le problème du décodage en métrique rang.

3.3 Code transposé

Les code LRPC utilisés dans le cryptosystème de McEliece[4] sont des codes $[n,k]$ sur $\mathbb{F}_{q^m}$ vérifiant $n \geqslant m$. Dans ce cas l'algorithme de Gaborit, Ruatta et Schrek est le plus efficace. La question se pose de savoir s'il est possible d'adapter cet algorithme dans le cas où $n < m$.
Comme les mots d'un code $[n,k]$ sur $\mathbb{F}_{q^m}$ s'identifient naturellement à des matrices $m \times n$, une idée est de transformer ce code en un code de longueur m sur $\mathbb{F}_{q^n}$ en transposant les matrices.

Définition 3.7 (Code transposé)**.** Soit $\mathcal{C}$ un code $[n,k]$ sur $\mathbb{F}_{q^m}$.
Soient $(\beta_1 \dots \beta_m)$ une base de $\mathbb{F}_{q^m}$ sur $\mathbb{F}_q$ et $(\gamma_1 \dots \gamma_n)$ une base de $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$.
Tout mot $c \in \mathcal{C}$ peut s'écrire sous forme matricielle :

$$(c_1, \dots, c_n) \leftrightarrow \begin{pmatrix} c_{11} & \dots & c_{1n} \\ \vdots & & \vdots \\ c_{m1} & \dots & c_{mn} \end{pmatrix}$$

où $c_i = \sum_{j=1}^{m} c_{ji}\beta_j$.
On définit $c^T \in \mathbb{F}_{q^n}^m$ par $c^T = (\sum_{i=1}^n c_{1i}\gamma_i, \dots, \sum_{i=1}^n c_{mi}\gamma_i)$.
Le code transposé de $\mathcal{C}$, noté $\mathcal{C}^T$,, est l'ensemble $\{c^T, c \in \mathcal{C}\}$

Par construction, l'application transposée est un isomorphisme de $\mathbb{F}_q$-espaces vectoriels qui préserve le rang des vecteurs. En revanche, elle ne conserve par linéarité par rapport à $\mathbb{F}_{q^m}$.
En effet soit $\mathcal{C}$ un code $[n,k]$ sur $\mathbb{F}_{q^m}$ et soit M_c la matrice associée à un mot $c \in \mathcal{C}$ selon une base $(\beta_1, \dots \beta_m)$ de $\mathbb{F}_{q^m}$ sur $\mathbb{F}_q$.
Soit $\alpha \in \mathbb{F}_{q^m}$, notons $\mu(\alpha)$ la matrice de l'isomorphisme d'espaces vectoriels $\mu(\alpha) : x \in \mathbb{F}_{q^m} \mapsto \alpha x \in \mathbb{F}_{q^m}$ dans la base des (β_i) :

$$\mu(\alpha) = \begin{array}{c} \begin{matrix} \alpha\beta_1 & \dots & \alpha\beta_m \end{matrix} \\ \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mm} \end{pmatrix} \end{array} \begin{matrix} \beta_1 \\ \vdots \\ \beta_m \end{matrix}$$

ie pour tout $i \in [\![1; m]\!], \alpha\beta_i = \sum_{j=1}^{m} a_{ji}\beta_j$

Matriciellement, $\mathcal{C}$ est stable par multiplication à gauche par les matrices des automorphismes $c \mapsto \alpha c$. L'action de transposer les mots du code inverse la multiplication matricielle, ce qui ôte la linéarité. Ainsi, il n'est plus possible d'obtenir simplement un facteur $r-1$ au lieu d'un facteur r dans le facteur exponentiel.

Dans le code transposé, le support E de l'erreur est un sous-espace vectoriel de dimension r de $\mathbb{F}_q^n$. Le même algorithme de décodage s'applique, si l'on connaît un espace vectoriel E' de dimension r' contenant E, on peut retrouver l'erreur à condition d'avoir plus d'équations de parité que d'inconnues.

La transposition ne change pas le nombre d'équations de parité qui vaut $(n-k)m$. En revanche, le nombre d'inconnues vaut mr' car le code est de longueur m, d'où $r' = n-k$.

La probabilité d'obtenir $E \subset E'$ vaut $\dfrac{\begin{bmatrix} r' \\ r \end{bmatrix}_q}{\begin{bmatrix} n \\ r \end{bmatrix}_q} \approx \dfrac{q^{r(r'-r)}}{q^{r(n-r)}} = q^{-r(n-r')}$.

En remplaçant r' par $n-k$, on obtient une complexité exponentielle en q^{rk}, ce qui correspond à la complexité de l'algorithme d'Ourivski et Johansson[9] sans l'utilisation de la linéarité par rapport à $\mathbb{F}_{q^m}$ qui permet de choisir un élément de la base du support de l'erreur et ainsi d'obtenir le $r-1$ dans l'exposant. A priori, il n'est pas facile d'obtenir cette amélioration dans le code transposé, c'est pourquoi nous avons décidé d'implémenter l'algorithme d'Ourivski et Johansson pour les codes de paramètre $m > n$.

4 Description des attaques

Dans cette partie, nous allons présenter deux attaques réalisées sur les codes LRPC doublement circulants. Des paramètres concrets pour leur utilisation dans le cryptosystème de McEliece ont été proposés dans cet article [4]. Les deux attaques utilisent pleinement la structure circulante de ces codes.

Avant de décrire ces attaques, nous allons rappeler quelques propriétés des matrices circulantes.

4.1 Matrices circulantes

Définition 4.1 (Matrice circulante)**.** Une matrice circulante d'ordre n est une matrice carrée $n \times n$ de la forme

$$\begin{pmatrix} m_1 & m_2 & \dots & m_n \\ m_n & m_1 & \dots & m_{n-1} \\ & \ddots & \ddots & \\ m_2 & m_3 & \dots & m_1 \end{pmatrix}$$

Formellement, soient $\mathbb{K}$ un corps et $M = (m_{ij})_{i \in \mathbb{Z}/n\mathbb{Z}, j \in \mathbb{Z}/n\mathbb{Z}} \in \mathbb{K}^{n \times n}$ une matrice carrée de taille n à coefficients dans $\mathbb{K}$.
M est circulante si et seulement si

$$\forall (i,j) \in [\![1;n]\!]^2, m_{ij} = m_{i-1,j-1}$$

Un résultat bien connu d'algèbre linéaire qui s'applique sur aux codes doublement circulants est qu'il existe une représentation polynomiale des matrices circulantes via un isomorphisme d'algèbre. Ce résultat fait l'objet de la proposition suivante :

Proposition 4.2. *Soient $\mathbb{K}$ un corps et $\mathcal{MC}_{\mathbb{K}}(n)$ l'ensemble des matrices circulantes de tailles n à coefficients dans $\mathbb{K}$.*
Soit R l'algèbre-quotient $\mathbb{K}[X]/(X^n - 1)$.

$$\begin{array}{rccc} \textit{L'application } \varphi & : & R & \to & \mathcal{MC}_{\mathbb{K}}(n) \\ & & \sum_{i=0}^{n-1} a_i X^i & \mapsto & \begin{pmatrix} a_0 & \dots & a_{n-1} \\ a_{n-1} & \ddots & a_{n-2} \\ a_1 & \dots & a_0 \end{pmatrix} \end{array}$$

est un isomorphisme de $\mathbb{K}$-algèbres.

Ainsi, $\mathcal{MC}_{\mathbb{K}}(n)$ est une sous-algèbre commutative de $\mathbb{K}^{n \times n}$.
Une matrice circulante M est inversible si et seulement si $\varphi^{-1}(M)$ est premier avec $X^n - 1$ et son inverse est aussi une matrice circulante.
La propriété suivante permet de définir le rang d'une matrice circulante et nous sera utile par la suite [6] :

Proposition 4.3. *Soit M une matrice circulante d'ordre n associé au polynôme P par l'automorphisme de la proposition 4.2.*
Le rang de M est $n-d$ où $d = \deg(P \wedge (X^n-1))$

Démonstration. Soit $D = P \wedge (X^n-1)$.
Soient $(q_0, \dots q_{n-1}) \in \mathbb{K}^n$ et $Q = \sum_{i=0}^{n-1} q_i X^i$.
Effectuons le division euclidienne de Q par $\frac{X^n-1}{D}$: $Q = \hat{Q}\frac{X^n-1}{D} + R$ avec $R(X) = \sum_{i=0}^{n-d-1} r_i X^i$.
Soit φ l'isomorphisme décrit à la proposition précédente. On a :

$$\begin{aligned} M\varphi(Q) &= \varphi(PQ) \\ &= \varphi(\frac{P}{D}\hat{Q}(X^n-1) + RP) \\ &= \varphi(RP) \end{aligned}$$

Donc $M(q_0, \dots q_{n-1})^T = M(r_0, \dots r_{n-d-1}, 0 \dots 0)^T$
d'où $\dim Im(M) \leqslant n-d$.

Supposons par l'absurde que le rang de M est $< n-d$. Cela implique que tout ensemble de $n-d$ lignes de M est lié.
En particulier, il existe $(\lambda_0 \dots \lambda_{n-d-1}) \in \mathbb{K}^{n-d}$, non tous nuls tels que

$$\sum_{i=0}^{n-d-1} \lambda_i L_i = 0 \text{ avec } L_i \text{ la i}^{\text{e}} \text{ ligne de } M$$

Soit $\Lambda(X) = \sum_{i=0}^{n-d-1} \lambda_i X^i$.
Comme M est circulante, $\varphi(\Lambda)M = \varphi(\Lambda P) = 0$
$\Rightarrow X^n - 1 | \Lambda P \Rightarrow \frac{X^n-1}{D} \big| \Lambda \frac{P}{D}$
Donc, d'après le théorème de Gauss, $\frac{X^n-1}{D} \big| \Lambda$, ce qui est absurde car $\deg(\Lambda) \leqslant n-d-1$.
D'où $\dim Im(M) = n-d$. □

4.2 Première attaque

Nous allons à présent introduire la notion de code replié par rapport à un polynôme, qui va nous permettre de mettre en place l'attaque. Cette notion a été pour la première fois présentée dans [11] et nous reprenons dans cette sous-section la présentation telle qu'elle est donnée dans ce papier.

Définition 4.4. Soient $\mathcal{C}$ un code doublement circulant de type $[n,k]$ sur $\mathbb{F}_{q^m}$ et G une matrice génératrice doublement circulante.
Soit R l'algèbre quotient $\mathbb{F}_{q^m}[X]/(X^k-1)$. D'après la proposition 4.2, il existe deux polynômes P et Q de R tels que $G = (\varphi(P)|\varphi(Q))$.
Soit $D \in \mathbb{F}_q[X]$ un diviseur de X^k-1 de degré d. Le code replié de $\mathcal{C}$ par D, noté $\mathcal{C}_{rep,D}$ (ou plus simplement $\mathcal{C}_{rep}$ quand il n'y a pas d'ambiguïté), est le

code engendré par la matrice $\varphi(D)G = \varphi(DP)|\varphi(DQ))$.
Considérons la division euclidienne de P et de Q par $\frac{X^k-1}{D}$:

$$\begin{cases} P &= \hat{P}\frac{X^k-1}{D} + R \\ Q &= \hat{Q}\frac{X^k-1}{D} + S \end{cases} \Rightarrow \begin{cases} PD &\equiv RD \mod X^k-1 \\ QD &\equiv SD \mod X^k-1 \end{cases}$$

Le code $\mathcal{C}_{rep}$ est ainsi engendré par la matrice $(\varphi(RD)|\varphi(SD))$ donc sa dimension est inférieure à $k-d$.
Dans le cas où P (ou Q) est inversible dans R, alors la matrice $(I_k, \varphi(P^{-1}Q))$ est aussi une matrice génératrice doublement circulante de $\mathcal{C}$ donc $(\varphi(D)|\varphi(DP^{-1}Q))$ engendre $\mathcal{C}_{rep}$. Dans ce cas, la dimension de $\mathcal{C}_{rep}$ vaut $k-d$ d'après la proposition 4.3.

Remarques :

1. Comme $D \in \mathbb{F}_q[X]$, les lignes de la matrice $\varphi(D)G$ s'obtiennent par combinaisons linéaires à coefficients dans $\mathbb{F}_q$ des lignes de G. Ainsi le sous-espace vectoriel de $\mathbb{F}_{q^m}$ engendré par les coefficients de $\varphi(D)G$ est inclus le sous-espace vectoriel engendré par les coefficients de G. Cette remarque est importante dans le cas des LRPC car le support des lignes de $\varphi(D)G$ est encore de petite dimension ; à la différence de la métrique de Hamming où le repliement d'un code ne conserve pas le support.
2. Comme G est la concaténation de deux matrices circulantes, on peut voir le repliement comme une action sur les colonnes de G. En effet, on a l'égalité suivante :

$$\varphi(D)G = G\begin{pmatrix} \varphi(D) & 0 \\ 0 & \varphi(D) \end{pmatrix}$$

   Ainsi certaines colonnes s'obtiennent par combinaison linéaire des autres, ce qui implique une dépendance linéaire entre les coordonnées des mots du code replié. Moralement, il existe un isomorphisme entre le code $\mathcal{C}_{rep}$ de type $[n, k-d]$ et un code de type $[n-2d, k-d]$ mais cette opération n'est a priori pas évidente.
   Il serait cependant très intéressant de chercher à projeter le code replié dans un code plus petit afin de réduire le coût des opérations d'algèbre linéaire.

Soit $\mathcal{C}$ un code LRPC $[n, k]$ sur $\mathbb{F}_{q^m}$ de poids d doublement circulant. Soient H une matrice doublement circulante de parité de $\mathcal{C}$ de poids d et F le sous-espace vectoriel de $\mathbb{F}_q^m$ de dimension d engendré par les coefficients de H.
Soient P et Q deux polynômes de $R = \mathbb{F}_{q^m}[X]/(X^k-1)$ tels que $H = (\varphi(P)|\varphi(Q))$.

En supposant que la matrice $\varphi(P)$ inversible, ce qui est très probable car ses coefficients appartiennent à $\mathbb{F}_{q^m}$, la matrice $H' = (I_k|\varphi(P^{-1}Q))$ est une autre matrice doublement circulante de parité du code. A priori, le poids de H' est m.
La clé secrète du cryptosystème est une description de H et la clé publique est une description de H'.

Pour appliquer l'algorithme de décodage des codes LRPC, il suffit de retrouver une matrice de parité de $\mathcal{C}$ de petit poids d à partir de H'. Pour cela, on va chercher des mots de petit poids d dans le code dual $\mathcal{C}^\perp$ dont H' est une matrice génératrice en utilisant une modification de l'algorithme 4.

Considérons le code replié $\mathcal{C}^\perp_{rep}$ par le polynôme $\frac{X^k-1}{X-1} = \sum_{i=0}^{k-1} X^i$. $\mathcal{C}^\perp_{rep}$ est le code $\mathcal{C}^\perp$ replié au maximum et sa dimension vaut 1. Une matrice-ligne génératrice de ce code est obtenue en sommant toutes les lignes de n'importe quelle matrice génératrice de $\mathcal{C}^\perp$
Appliqué à la matrice H, ce repliement permet d'obtenir la matrice $(\underbrace{\beta,\ldots,\beta}_{k\ fois},\underbrace{\gamma,\ldots,\gamma}_{k\ fois})$ comme matrice génératrice de $\mathcal{C}^\perp_{rep}$, avec $\beta,\gamma \in F$.
Appliqué à la matrice H', on obtient la matrice $(1,\ldots,1,\alpha,\ldots,\alpha)$.
Dans ce cas précis, il est évident que $\mathcal{C}^\perp_{rep}$ est isomorphe au code $[2,1]$ engendré par $(1,\alpha)$. Comme le mot (β,γ) appartient à ce code, $1,\alpha \in \beta^{-1}F$.
Ainsi, ce code replié permet d'obtenir deux vecteurs d'une base d'un multiple F' de F.

Comme dans l'algorithme 4, on cherche un espace vectoriel E' de dimension r' contenant F'. En exprimant les coordonnées d'un mot dans E', on peut retrouver un mot de poids d donc reconstruire une matrice de parité de poids d.
Le nombre d'inconnues est $nr' = 2kr'$ et le nombre d'équations de parité est $m(n-k) = mk$, d'où $r' = \lfloor \frac{m}{2} \rfloor$.
Comme on connaît à l'avance deux éléments d'une base de F', la probabilité d'avoir $F' \subset E'$ est :

$$\frac{\begin{bmatrix} r' \\ d-2 \end{bmatrix}_q}{\begin{bmatrix} m \\ d-2 \end{bmatrix}_q} \approx q^{-(d-2)(m-r')} = q^{-(d-2)\lceil \frac{m}{2} \rceil}$$

Données : $H' = (I_k|M)$: matrice de parité publique de $\mathcal{C}$
d : poids de $\mathcal{C}$
Sorties : H : matrice de parité de poids d de $\mathcal{C}$

début
  $r' \leftarrow \lfloor \frac{m}{2} \rfloor$;
  calculer une matrice de parité G du code $\mathcal{C}^\perp$;
  /* G est une matrice génératrice de $\mathcal{C}$ */
  $\alpha \leftarrow \sum_{i=1}^{k} m_i$ où $(m_1 \dots m_k)$ est la première ligne de M;
  bool $\leftarrow$ TRUE;
  tant que *bool* **faire**
    choisir aléatoirement E' de dimension r' contenant 1 et α;
    calculer une base $\{E'_1 \dots E'_{r'}\}$ de E';
    /* les inconnues du système sont les coordonnées d'un mot e dans la base (E'_j) */
    pour chaque $i \in [\![1; n]\!]$ **faire**
      exprimer e_i dans la base des (E'_j) ;
      /* $e_i = \sum_{j=1}^{r'} e_{ij} E'_j$ */
    finpour
    résoudre le système $Ge^T = 0$;
    si *le poids de e est d* **alors**
      bool $\leftarrow$ FALSE;
    finsi
  fintq
  /* à ce stade, on connaît un mot e de poids d du code $\mathcal{C}^\perp$ */
  calculer une base $\{E_1 \dots E_d\}$ du support E de e;
  résoudre le système $Gy^T = 0$ en exprimant les coordonnées de y dans la base (E_j) jusqu'à obtenir k mots $\{y_1 \dots y_k\}$ de poids d indépendants;
  $H \leftarrow \begin{pmatrix} y_1 \\ \vdots \\ y_k \end{pmatrix}$;
  retourner H
fin

Algorithme 6 : calcul d'une matrice de parité de petit poids d'un code LRPC

4.3 Amélioration de l'attaque

Pour améliorer cette attaque sur les codes LRPC doublement circulants, nous allons utiliser le repliement pour diminuer la dimension du code. Comme expliqué dans la remarque page 25, le code dual replié d'un code LRPC possède également une matrice génératrice dont les coefficients engendrent F. Il s'agit donc de retrouver des mots de poids d dans ce code pour reconstruire un multiple F' de F. Comme la dimension du code replié est plus petite, cette recherche est plus rapide.

De plus, comme le repliement agit aussi sur les colonnes de la matrice génératrice donc induit une redondance dans les coordonnées, on peut être amené à rechercher des mots de petits poids dans un code de longueur inférieure au degré de l'extension de $\mathbb{F}_{q^m}$. C'est pourquoi nous avons choisi d'utiliser l'algorithme d'Ourivski et Johansson [9] pour cette recherche.

4.3.1 Description de l'algorithme

Soit $\mathcal{C}$ un code $[n,k]$ sur $\mathbb{F}_{q^m}$. On suppose que $\mathcal{C}$ admet une matrice génératrice sous forme systématique : $G = (I_k|R)$.
Soit $c \in \mathcal{C}$ un mot de poids d. Écrivons $c = (c_1, c_2)$ avec c_1 de longueur k et c_2 de longueur $n-k$. On a donc l'égalité :

$$c_2 = c_1 R \tag{1}$$

Comme c est par hypothèse de rang d, il peut s'écrire sous la forme :

$$c = (x_0, \dots, x_{d-1}) \begin{pmatrix} a_{01} & \dots & a_{0k} & a_{0,k+1} & \dots & a_{0n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{d-1,1} & \dots & a_{d-1,k} & a_{d-1,k+1} & \dots & a_{d-1,n} \end{pmatrix} = xA$$

où les éléments $x_0, \dots x_{d-1}$ forment une base du support de c et où A est une matrice $d \times n$ à coefficients dans $\mathbb{F}_q$ de rang d.

Dans le cas où on connaît un ou plusieurs éléments du support (par linéarité par rapport à $\mathbb{F}_{q^m}$, on peut toujours supposer que 1 appartient au support), on peut remplacer les x_i par leur valeur. Dans le cas des codes LRPC doublement circulants, on connaît deux éléments, 1 et α, du support d'après la première attaque.

Avec ces notations, on peut écrire $\begin{cases} c_1 = xA_1 \\ c_2 = xA_2 \end{cases}$ où A_1 est constituée des k premières colonnes de A et A_2 des $(n-k)$ dernières.
Notons r_j la $(j-k)^{\text{e}}$ colonne de R, avec $k+1 \leqslant j \leqslant n$. D'après (1), on a :

$$xA_1 r_j = x \begin{pmatrix} a_{0j} \\ \vdots \\ a_{d-1,j} \end{pmatrix} \tag{2}$$

Cette égalité peut se réécrire

$$x \begin{pmatrix} & a_{0j} \\ A_1 & \vdots \\ & a_{d-1,j} \end{pmatrix} \begin{pmatrix} r_j \\ -1 \end{pmatrix} = 0 \tag{3}$$

ce qu'on écrira de manière plus condensée $xA_1(j)\tilde{r}_j = 0$ avec $A_1(j)$ la concaténation de A_1 et de la j^{e} colonne de A et $\tilde{r}_j$ le vecteur colonne $\binom{r_j}{-1}$
On obtient donc un système de $n-k$ équations sur $\mathbb{F}_{q^m}$ d'inconnues $x_2, \dots x_{d-1}$ et A.
Pour transcrire ce système dans $\mathbb{F}_q$, on se munit d'une base $(\beta_1 \dots \beta_m)$ de $\mathbb{F}_{q^m}$ sur $\mathbb{F}_q$. On peut ainsi définir l'isomorphisme de $\mathbb{F}_q$ espaces vectoriels

$$\begin{array}{rccc} p & : & \mathbb{F}_{q^m} & \to & \mathbb{F}_q^m \\ & & y = \sum_{i=1}^m y_i\beta_i & \mapsto & (y_1, \dots, y_m) \end{array}$$

Soit u et v deux éléments de $\mathbb{F}_{q^m}$ et $w = uv$. On peut montrer qu'il existe m matrices carrées Δ_i de tailles m inversibles telles que :

$$\forall i \in [\![1; m]\!], p(w)_i = p(u)\Delta_i p(v)^T \tag{4}$$

Ces matrices ne dépendent que de la base (β_i) choisie.

À la colonne r_j de R, on fait correspondre une matrice Υ_j de taille $k \times m$ telle que $p(r_{ij})$ soit égal à la i^{e} ligne de Υ_j :

$$\begin{pmatrix} r_{1j} \\ \vdots \\ r_{kj} \end{pmatrix} \leftrightarrow \begin{pmatrix} \rho_{11}^j & \dots & \rho_{1m}^j \\ \vdots & & \vdots \\ \rho_{k,1}^j & \dots & \rho_{km}^j \end{pmatrix}$$

Soit λ la représentation de $-1 \in \mathbb{F}_{q^m}$
Notons $A_1(j)\tilde{r}_j = (b_{0j}, \dots, b_{d-1,j})^T$. On a alors :

$$b_{ij} = (\Upsilon_j \lambda^T) \begin{pmatrix} a_{i1} \\ \vdots \\ a_{ik} \\ a_{ij} \end{pmatrix} \text{ pour tout } i \in [\![0; d-1]\!]$$

Notons $p(x_i) = (\gamma_{1i} \dots \gamma_{mi})$ et $p(\alpha) = (\alpha_1 \dots \alpha_m)$.
Notons $p(x_i c_{ij}) = (\sigma_1(i,j), \dots, \sigma_m(i,j))$. D'après (4), on a :

$$\sigma_l(i,j) = (\gamma_{1i} \dots \gamma_{mi})\Delta_l(\Upsilon_j \lambda^T) \begin{pmatrix} a_{i1} \\ \vdots \\ a_{ik} \\ a_{ij} \end{pmatrix}, \text{ avec } l \in [\![1; m]\!], i \in [\![2, d-1]\!] \tag{5}$$

Comme $x_0 = 1$, on a :

$$\sigma_l(0,j) = \mu_l(j) \begin{pmatrix} a_{01} \\ \vdots \\ a_{0k} \\ a_{0j} \end{pmatrix} \tag{6}$$

D'après (3), on a $\sum_{i=0}^{d-1} x_i c_{ij} = 0$ d'où d'après (5) et (6) :

$$\sum_{i=2}^{d-1} (\gamma_{1i} \dots \gamma_{mi})(\Upsilon_j \lambda^T) \begin{pmatrix} a_{i1} \\ \vdots \\ a_{ik} \\ a_{ij} \end{pmatrix} + (\alpha_1 \dots \alpha_m)\Delta_l(\Upsilon_j \lambda^T) \begin{pmatrix} a_{i1} \\ \vdots \\ a_{ik} \\ a_{ij} \end{pmatrix} + \mu_l(j) \begin{pmatrix} a_{01} \\ \vdots \\ a_{0k} \\ a_{0j} \end{pmatrix} = 0$$

pour tout $j \in [\![k+1; n]\!]$ et $l \in [\![1; m]\!]$ On a donc un système quadratique de $(n-k)m$ équations à $(d-2)m + dn$ inconnues (les x_i plus les coefficients de A).

Pour résoudre ce système, on va deviner les coefficients de A qui interviennent dans la partie non linéaire. Comme il est suffisant de calculer c_1 pour retrouver c, il n'est pas nécessaire de deviner ces coefficients pour tout j, on suppose que l'on devine ces coefficients pour seulement N valeurs de j, soit $k(d-2)$ (les coefficients de A_1 sauf les deux premières lignes qui sont des inconnues) plus $N(d-2)$ (les coefficients des N colonnes tronquées) valeurs à deviner. Au final, le système possède $2(k+N) + (d-2)m$ inconnues pour mN équations dans $\mathbb{F}_q$. On peut résoudre ce système à condition d'avoir $Nm \geqslant 2(k+N) + (d-2)m$, soit :

$$N \geqslant \left\lceil \frac{2k + (d-2)m}{m-2} \right\rceil = d - 2 + \left\lceil \frac{2(k+d-2)}{m-2} \right\rceil (*)$$

On peut diminuer le nombres de valeurs à deviner en constatant que certaines valeurs conduisent à calculer le même mot. En effet soit B une matrice inversible de $\mathbb{F}_q^{d \times d}$.
c peut s'écrire $(1, \alpha, x_2, \dots, x_{d-1})BB^{-1}A$. Cette matrice B donne d'autres valeurs pour $x_2, \dots x_{d-1}$ mais permet de calculer le même c. Ainsi, A peut être choisie parmi les matrices échelonnées. Il y a

$$\frac{(q^{k+N} - 1) \dots (Q^{k+N} - q^{d-3})}{(q^{d-2} - 1) \dots (q^{d-2} - q^{d-3})} \leqslant q^{(d-2)(k+N+2-d)+2}$$

matrices de cette forme d'après [9]. En prenant en compte le coût de l'algèbre linéaire pour la résolution du système, on obtient une complexité en

$$\mathcal{O}\big((mN)^3 q^{(d-2)(k+N+2-d)+2}\big)$$

En remplaçant N par (*), la complexité de l'algorithme est de l'ordre de

$$\mathcal{O}\big(m^3\Big(d-2+\Big\lceil\frac{2(k+d-2)}{m-2}\Big\rceil\Big)^3 q^{(d-2)(k+\lceil\frac{2(k+d-2)}{m-2}\rceil)+2}\big)$$

Dans le cas des codes LRPC replié, on a très souvent m de l'ordre de $2(k+d-2)$ puisque d est choisi petit et que le repliement permet de diminuer k.

4.3.2 Exemples de paramètres

L'article [4] présente trois paramètres de code LRPC que nous présentons ci-dessous. La colonne sécurité correspond à la sécurité du cryptosystème donnée dans l'article.

n	k	m	q	d	r	sécurité
74	37	41	2	4	4	80
94	47	47	2	5	5	128
68	34	23	2^4	4	4	100

Pour attaquer ces codes grâce au repliement, il faut d'abord factoriser le polynôme X^k-1, k étant la dimension du code. Plus ce polynôme possède de facteurs, plus le nombre de repliements possibles est grand, donc plus il est fragile face à cette attaque.
Voici la factorisation des trois polynôme. On ne donne pas la valeur des différents facteurs dans tous les cas, puisque c'est plus le degré et le nombre de facteurs qui nous intéressent.

— $X^{37}-1=(X-1)\sum_{i=0}^{36}X^i$
— $X^{47}-1=(X-1)PQ$ avec $\deg(P)=\deg(Q)=23$
— $X^{34}-1=(X-1)^2(P_1\dots P_8)^2$ avec pour tout $i\in[\![1;8]\!],\deg(P_i)=2$

Comme on peut le voir, le repliement n'apporte pas d'attaque efficace pour le premier jeu de paramètres.

Le deuxième cas est plus intéressant car on peut se ramener à un code replié de dimension 23 en choisissant $D=(X-1)P$ ou $(X-1)Q$ pour diviseur de X^k-1.
Ici $d=5$ et $\lceil\frac{2(k+d-2)}{m-2}\rceil=2$.
Avec la version améliorée de l'algorithme d'Ourivski et Johansson, en remplaçant les paramètres par leur valeur, la complexité est de l'ordre de

$$\mathcal{O}\big(47^3 5^3 2^{3(23+2)+2}\big)\leqslant 2^{101}$$

en utilisant l'inégalité $47\times 5=235<2^8$. Le gain en complexité est donc très important. À titre de comparaison, la complexité obtenue dans [11] pour

ce code est 2^{117}. La différence provient du fait que l'attaque n'utilise pas le repliement total par rapport à $\sum_{i=0}^{k-1} X^i$ qui permet d'obtenir deux vecteurs d'un multiple du support.

Le dernier cas est le plus intéressant. Le polynôme $X^{34} - 1$ se factorise en 8 polynômes de degré 2 et un polynôme de degré 1, tous de valuation 2. Cela permet de choisir n'importe quelle valeur pour la dimension du code replié.
Pour $k \leqslant 8$, on a $\lceil \frac{2(k+d-2)}{m-2} \rceil = 1$. Le tableau suivant résume la complexité de l'attaque en fonction de différents k :

k	complexité
8	2^{98}
6	2^{82}
4	2^{66}

Dans les différents cas, la distance de Gilbert-Varshamov en métrique rang est supérieure à 4, ce qui assure que le décodage est unique. Pour $k \leqslant 5$, la complexité passe sous la barre des 2^{80}, ces paramètres ne sont donc pas sûrs.

5 Conclusion

Ce stage de six mois m'a permis de découvrir le monde de la recherche et le métier de chercheur. Cela a été une expérience formatrice pour moi.

Durant ce stage, nous avons cherché comment appliquer et comment traduire les différentes avancées dans le décodage de codes en métrique de Hamming au décodage de codes en métrique rang. Cela a permis de proposer la notion de complémentaire du support et de calculer la probabilité que la dimension de l'intersection de deux supports vaille a. Bien que nous n'ayons pas pu trouver d'algorithmes améliorant la complexité du décodage en métrique rang, cela jette les bases d'un futur algorithme et donne une voie pour orienter les recherches.

Dans une seconde partie, je me suis intéressé à une famille de codes, les codes LRPC[4], qui ont des applications en cryptologie à clé publique. Une attaque structurelle sur la sous-famille des codes LRPC doublement circulants a été proposée, qui parvient à casser certains paramètres. Il est donc nécessaires de tenir compte de cette attaque pour la conception de cryptosystèmes basés sur ces codes.

Les perspectives qu'ouvrent ce stage sont l'amélioration du décodage des code en métrique rang, l'implémentation de l'attaque sur les codes LRPC doublement circulants et étudier une possible généralisation à tous les paramètres de ces codes.

Références

[1] Ravzan Barbelescu, Pierre Gaudry, Antoine Joux, and Emmanuel Thomé. *A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic.* In *Advances in Cryptology–EUROCRYPT 2014*, pages 1–16. Springer, 2014.

[2] Elwyn R Berlekamp and Robert J and McEliece. *On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory*, 24(3) :384–386, 1978.

[3] Florent Chabaud and Jacques Stren. *The cryptographic security of the syndrom decoding problem for rank distance codes.* In *Advances in Cryptology-ASIACRYPT'96*, pages 368–381. Springer, 1996.

[4] Philippe Gaborit, Gaétan Murat, Olivier Ruatta, and Gilles Zémor. *Low Rank Parity Check codes and their application to cryptography.* In *The preproceedings of Workshop on Codes and Cryptography (WCC)*, pages 167–179, 2013.

[5] Philippe Gaborit, Olivier Ruatta, and Julien Schrek. *On the complexity of the Rank Syndrome Decoding problem. arXiv preprint arXiv :1301.1026*, 2013.

[6] AW Ingleton. The rank of circulant matrices. *Journal of the London Mathematical Society*, 1(4) :445–460, 1956.

[7] Harald Lidl, Rudolfand Niederreiter. *Finite Fields.* Cambridge University Press, Second edition, 1997.

[8] Robert J McEliece. *A Public-Key Cryptosystem Based On Algebraic Coding Theory. DSN Progress Report*, 42(44) :114–116, janvier février 1978.

[9] Alexei V Ourivski and Thomas Johansson. *New techniques for decoding codes in the rank metric and its cryptography applications. Problems of Information Transmission*, 38(3) :237–246, 2002.

[10] Gaborit Philippe and Zemor Gilles. On the hardness of the decoding and the minimum distance problems for rank codes. *arXiv preprint arXiv :1404.3482*, 2014.

[11] Loidreau Pierre. On cellular code and their cryptographic applications. Communication personnelle.

[12] Nicolas Sendrier. *Finding the permutation between equivalent codes : the support splitting algorithm. IEEE Transaction on Information Theory*, 46(4) :1193–1203, July 2000.